Kuang et al. Journal of Cloud Computing: Advances, Systems and Applications (2020) 9:10
https://doi.org/10.1186/s13677-020-0158-3

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH Open Access

Providing privacy preserving in next POI recommendation for Mobile edge computing

Li Kuang[1], Shenmei Tu[1], Yangqi Zhang[1] and Xiaoxian Yang[2*]

Abstract

Point of interest (POI) recommendation can benefit users and merchants. It is a very important and popular service in modern life. In this paper, we aim to study the next new POI recommendation problem with the consideration of privacy preserving in edge computing. The challenge lies in capturing the transition patterns between POIs precisely and meanwhile protecting users' location. In this paper, first, we propose to model users' check-in sequences with their latent states based on HMM, and EM algorithm is used to estimate the parameters of the model. Second, we propose to protect users' location information by a weighted noise injection method. Third, we predict users' next movement according to his current location based on Forward algorithm. Experimental results on two large-scale LBSNs datasets show that our proposed model without noise injection can achieve better recommendation accuracy than several state-of-the-art techniques, and the proposed weighted noise injection approach can achieve better performance on privacy preserving than traditional one with a little cost on accuracy.

Keywords: HMM, Sequential transition patterns, Latent state, Privacy preserving, POI recommendation

Introduction

Mobile edge computing (MEC) is a "hardware + software" system that reduces network operations and service delivery delays by providing IT service environments and cloud computing capabilities on the edge of the mobile network. Its technical features include "proximity, location awareness, high bandwidth and low latency". MEC can provide users with local video, AR/VR, user positioning, video QoS optimization, video monitoring, traffic analysis and other services [1–3]. A typical application of MEC is location-based social media networks, where users can check-in and discuss their experiences. The check-in data of users can be used to study the life patterns for improving the performance of POI recommendation. POI recommendation provides recommendations of places to users, which is a popular service in LBSN.

Traditional POI recommendation usually recommends POIs to users according to the location, time, others' comments or their friends' interest. Being different from traditional POI recommendation, the next POI recommendation additionally considers users' check-in sequential transitions. Different kinds of people may exhibit different sequential transition patterns. For example, for students, after studying in the classroom, they may prefer to participating in intense outdoor activities or having meals in the canteen. While for workers in workdays, after working at offices, they may prefer to going shopping or going home. Meanwhile, we observe that users always visit new POIs that they have never visited before, so it is meaningful to mine the sequential transition patterns for suggesting new next POIs to users.

However, the computing power of cloud data centers has been centralized at the edge of the network by MEC. On one hand, MEC infrastructure is usually deployed at the edge of the network, such as wireless base stations, making it more vulnerable to unsafe environments. On the other hand, MEC will adopt open application programming interface (API), open network function virtualization (NFV) and other technologies, and the introduction of openness will easily expose MEC to external attackers [4, 5]. MEC services are faced with certain security risks due to limited resources and capabilities of edge facilities. Therefore, when

* Correspondence: xxyang@sspu.edu.cn
[2]School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China
Full list of author information is available at the end of the article

users check in on LBSNs, specific location information is needed to submit to the edge server, the trajectory involved in users' check-in data may reveal users' privacy, such as home address, living habits, and social relations. If such information is leaked to malicious attackers, the user will be exposed to a serious threat. Therefore, it is necessary to protect users' current location when recommending POIs to users.

On one side, in order to recommend POIs with high accuracy, many recent studies have been conducted, including temporal influence enhanced POI recommendation, geographical influence enhanced POI recommendation, content-aware POI recommendation, and social influence enhanced POI recommendation. However, the sequential transition patterns between successive check-ins has not yet well-studied. The next POI that a user will visit may be correlated with the former POI where he has been, and a large number of sequential transitions between POIs may exhibit a certain transition patterns for specific people, while such patterns may change with the time, therefore, it is a challenge to improve the performance of next POI recommendation by capturing such sequential transition patterns precisely.

On the other side, researchers have proposed many privacy preserving methods to solve the problem of privacy disclosure, including K-anonymity, cryptography and obfuscation. K-anonymity is easy to implement; however, the original location is still easy to be recognized. Cryptography has a strong protection ability; however, it needs a large amount of computational cost so that it is hard to apply. Obfuscation is achieved by injecting noise into the original data; however, many existing obfuscation methods always result in low data availability. Therefore, it is also a challenge to find a tradeoff between recommendation accuracy and privacy protection by designing a sound way of noise perturbation.

In this paper, we aim to achieve the satisfying performance of next POI recommendation and meanwhile prevent user's location information from being leaked in edge computing. First, we propose to use Hidden Markov Model (HMM) to capture user's latent state and transition patterns from the successive check-in data, and the model parameters are estimated by EM algorithm. Second, we propose to provide weighted noise perturbation of users' current locations based on the distance between user's current location and the nearest center. Finally, we propose to use the Forward algorithm for predicting the probability of users' next movement. We evaluate our proposed approach by extensive experiments on large-scale LBSN datasets and the results demonstrate that our proposed approach can not only effectively prevent the leakage of user's location privacy, but also have a satisfying accuracy of POI recommendation.

The remainder of the paper is organized as follows. Related work is discussed in Section "Related Work". The proposed approach is illustrated in Section "Proposed Method". The experimental results are presented in Section "Experiment". And finally, the conclusions and future work are given in Section "Conclusion and Future Work".

Related work

Many researchers pay attention to the development of POI recommendation because of the variety of applications in real life. POI recommendation was studied on the check-in behavior of many users [6]. The research of POI recommendations can be divided into four kinds: **1) Temporal influence enhanced POI recommendation**, which mainly investigates the influence of time on check-in data. Yuan et al. establish a new model based on the fact that users visit different places at different times [7]. Yin et al. study successive check-ins based on users' preference as well as the temporal context [8]. Although these works considered the temporal information, the temporal relationship between successive check-ins has not been well studied. **2) Geographical influence enhanced POI recommendation**, which studies "the phenomenon of geographical clustering" of check-ins for improving POI recommendation accuracy. Liu et al. propose a framework by combining Bayesian non-negative matrix factorization with geographical influence to achieve a good performance of POI recommendation [9]. Ye et al. explore the geographical influence of check-ins and propose a model that incorporates three factors: geographical impact, social impact and user preference [10]. **3) Content-aware POI recommendation**, which detect users' current locations by analyzing their published tweets and rank POIs by analyzing user's comments on them. Chen et al. propose a method to detect user preference from text content and establish a relationship between user preference and POI [11]. Gao et al. integrate user sentiment information and POI-related contents into POI recommendation system [12]. Due to the fact that many text contents in LBSN are vague, semantic analysis is a difficult research problem for POI recommendation. **4) Social influence enhanced POI recommendation**, researchers study social impact because many people believe that friends often have many common preferences in locations. By studying social impact, the quality of recommendation could be enhanced. However, there are also studies showing that many friends have little common interests in terms of POI visits [13] . And E. Cho et al. report their findings that social relationships have a greater impact on long distance journey [14].

The next new POI recommendation is a newly emerging issue and even challenging. There exist only a few works on this research issue in the literature. Feng et al.

propose a personalized ranking metric embedding method to study users' check-ins for next new POI recommendation [15]. Cheng et al. establish a model for exploring the order of location visits [16]. Feng et al. propose a POI2vec algorithm to jointly learn the latent representation for users and POIs, and then capture users' preference and POI sequence to improve the accuracy [17]. Oppokhonov et al. find the user's current location and then recommends new POIs based on collaborative filtering [18]. Since users' preferences are changing with the time going on and the check-in data exhibits users' latent states, although the above studies have considered the sequential influence, these studies have not yet well studied the latent states and users' transition patterns hidden in the check-in data.

In order to relieve the privacy disclosure in POI recommendation, researchers have proposed many traditional privacy protecting methods, which can roughly divided into three kinds: **1) K-anonymity** is a general privacy preserving method, which makes it possible for attackers to identify the target user with 1/k probability by ensuring the targeted user is indistinguishable from k set of users [19]. Gedik et al. study a customized framework to support k-anonymity with variable k to meet the personalized requirements of location privacy preserving [20]. However, the location of the target user can still be easily recognized. **2) Cryptography** provides a strong privacy preserving based on encryption and decryption mechanisms [21]. The data is firstly encrypted using some algorithms at the client and then transferred over the network. Liu et al. use partially homomorphic encryption technology to prevent users' location information from leaking [22]. However, due to the large calculation and the complexity of security protocols, this method is difficult to apply widely. **3) Obfuscation** uses a random value to replace a certain percentage of the user's information for achieving the purpose of privacy protection [23]. Ardagna et al. use different obfuscation operators for protecting users' location information, which can be used individually or in combination [24]. Polat et al. propose a randomized perturbation, where randomness noises following a specific distribution are injected into the original data [25].

Although the existing methods can provide privacy protection to a certain extent, the accuracy of POI recommendation has been decreased due to the decrease of data availability, in addition, it is hard to determine the magnitude of noise to be injected and the proportion of information to be replaced. So achieving the balance between location privacy protection and data availability is a difficult task.

Proposed method

Problem definition

$U = \{u_1, u_2, \ldots, u_s\}$ denotes a set of LBSN users, and I denotes POI, which is geo-coded by {longitude, latitude, category}. The track of user u_i before time t is represented by $O_{u_i} =< I_{u_i1}, I_{u_i2}, \ldots, I_{u_it} >$ .Given the tracks of all users $\{O_{u_1}, O_{u_2}, \ldots, O_{u_s}\}$, our goal is to precisely recommend top-k POIs for users' next movement at time $t + 1$ and meanwhile effectively protect the location privacy of users.

Approach overview

Figure 1 shows that our proposed method mainly includes three steps: constructing sequential model, injecting noise and constructing prediction model. 1) In the first step, we adopt HMM to model users' check-in data as POI sequential transitions. It is meaningful to capture POI sequential transition since a user's subsequent movement behavior is highly influenced by the

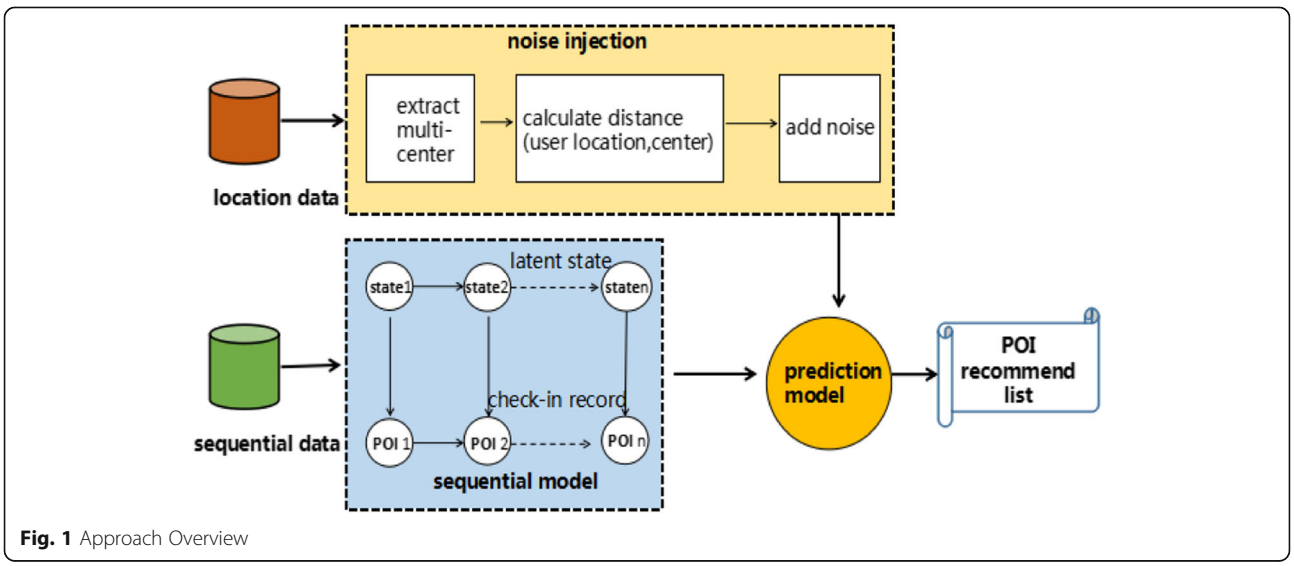

Fig. 1 Approach Overview

previously visited locations. 2) In the second step, we extract check-in centers from the check-in data first, and then we perturb users' current location by injecting Laplace noise according to the distance between users' location and the nearest check-in center. The improved noise injection way prevents user's location information from leaking to malicious attackers. 3) In the third step, we predict POIs for users' next movement according to the perturbed user's location by the forward algorithm.

Constructing sequential model

The base of sequential model is motivated by the following behavior in modern life:

(1) The check-in location is closely related to people's current state. For instance, people may keep the state of "eating breakfast in the morning" or "writing code in the afternoon". In the former state, people may check-in at coffeehouse, but in the latter state, people may check-in at the office.

(2) The change of state highly depends on the previous state. The state is a description of a series of behaviors over a period of time. People's state will not change suddenly because many people have a regular lifestyle. In other words, over a period of time, the state changing rules for regular users will remain stable. For example, after eating breakfast, people is most likely to go to work. Whether she is going to do architectural design on a business trip or at the office, she is going to do her work. Therefore, it is appropriate to apply a deterministic finite automaton for describing people's state changing rules [26].

The above phenomena are the base of the proposed sequential model. Hidden Markov Model (HMM) is a doubly stochastic process with an underlying stochastic process that is hidden, but can only be observed through another set of stochastic processes that produce the sequence of observed symbols, it manifests a rather sequentially changing process, the property of the process is usually held pretty steadily, except for minor fluctuations, for a certain period of time, and then, at certain instances, change to another property [27]. This property in HMM is very similar to users' states in real life. Since the lifestyle of many people is regular, and the changing rule of user's states will be stable over a period of time. Therefore, it is appropriate to adopt HMM to model users' check-in data for POI sequential transitions.

Modeling the check-in data by HMM

Firstly, to simplify the expression, we provide the parameter definitions as shown in Table 1.

Estimating parameters by EM algorithm

The main task of this section is to estimate three important parameters: the initial state distribution π, the state transition probability distribution T, and the preference probability distribution P.

Table 1 Parameters definitions

N	number of user states in the model
q_i	The i_{th} user's state
Q_u	$Q_u = \{q_1, q_2, \ldots q_N\}$, the state set of user u
M	number of POIs visited by user.
$t_{i,j}$	the probability that the state q_i transforms to the state q_j
T	$T = \{t_{i,j}\}_{N*N}$, state transition probability distribution
L	length of the POI sequence
O	$O = \{l_1, l_2, \ldots, l_M\}$, the POI sequential data
$p_{i,j}$	the probability that user visit POI l_j when the state is q_i
P	$p = \{p_{i,j}\}_{N*M}$, preference probability distribution
π	the probability of initial state

Generally, it is considered that the POI sequential record O is the most likely time series when given the three important parameters, so people can maximize the probability $\Pr(O_u/u)$ to get the parameters. The Expectation Maximization (EM) algorithm is always used to solve optimization problems. We represent $\alpha_t(i)$ as a forward vector, this vector denotes the probability of generating a POI sequence $<I_1, I_2, \ldots, I_t>$ in the state q_i at time t. Similarly,we represent $\beta_t(i)$ as a backward vector, this vector denotes the probability of generating a POI sequence $<I_{t+1}, \ldots, I_L>$ when current user state is q_i at time t. So we can formulate forward vector and backward vector recursively:

$$\alpha_t(j) = \left(\sum_{i=1}^{N} \alpha_{t-1}(i) T_{i,j}\right) P_{j,I_t} \tag{1}$$

$$\beta_t(i) = \sum_{j=1}^{N} T_{i,j} P_{j,I_{t+1}} \beta_{t+1}(j) \tag{2}$$

$\xi_t(i,j)$ is the probability that user state q_i transforms to q_j, which can be denoted with α and β:

$$\begin{aligned} \xi_t(i,j) &= \Pr\left(userstate_t = q_i, userstate_{t+1} = q_j / O_u, \mu\right) \\ &= \frac{\Pr\left(userstate_t = q_i, userstate_{t+1} = q_j / Ou, \mu\right)}{\Pr(O_u/\mu)} \\ &= \frac{a_t(i) T_{i,j} P_{j,I_{t+1}} \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} a_t(i) T_{i,j} I_{t+1} \beta_{t+1}(j)} \end{aligned} \tag{3}$$

And $\gamma_t(i)$ represent that when user in the t_{th} step, the user state is q_i:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{4}$$

The three core parameters in POI recommendation can be denoted by $\xi_t(i,j)$ and $\gamma_t(i)$ as follows:

$$\Pi_i := \Upsilon_t(i) \tag{5}$$

$$\mathrm{T}_{i,j} := \frac{\sum_{t=1}^{L}\xi_t(i,j)}{\sum_{t=1}^{L}\gamma_t(i)} \tag{6}$$

$$\mathrm{P}_{i,j} := \frac{\sum_{t=1}^{L}\gamma_t(i)*\delta(I_t,I_j)}{\sum_{t=1}^{L}\Upsilon_t(i)} \tag{7}$$

The parameters of the next iteration are obtained by maximizing the expectations of $\mu = \{\pi, T, P\}$. The iteration ends when μ converges. Hence, μ can be calculated by EM algorithm.

Perturbing user's location with noise injection

If users send their specific information to the LBSN, their private information may be revealed. We propose a check-in centers distance weighted noise injection algorithm to protect user's locations information in this paper. In order to prevent users' location information from being leaked, we inject different quantity of noise into the latitude and longitude of the user's location based on the distance between user's current location and their nearest check-in center. Then, we use the perturbed data for the next POI recommendation.

Noise injection is an important method of differential privacy mechanism. This method is to inject noise into the initial data and achieve the goal of privacy preserving by perturbing the object data. Laplace mechanism is a method for differential privacy protection [28]. The Laplace mechanism can be used for numerical datasets, it ensures the validity of ε-differential privacy by injecting noise into the original dataset D with Laplace distribution:

$$\mathrm{M(D)} = \mathrm{f(D)} + \mathrm{Y} \tag{8}$$

, where $Y \sim Lap(\Delta f/\varepsilon)$. However, such method of noise injection injects the same level of noise into the location data, resulting in low data availability and large impact of noise on the results.

An important feature of check-in data is that it obeys the Gaussian distribution around several check-in centers [29], as Fig. 2 shows. Another important feature is that most check-ins occur in nearby areas, and the probability that a user will visit a POI is in inverse proportion to the distance between users' location and the nearest check-in center [15]. Based on the two important features of check-in record, we propose a check-in center distance weighted noise injection method to protect users' location. Firstly, we find out all the check-in centers according to the greedy clustering algorithm proposed by Chen, which scans from the most visited POIs, and then merges POIs with a distance no larger than d kilometers into one region, if the ratio of the total number of check-in behavior in the area to the total number of check-in behavior of the user is greater than the threshold θ, then the check-in positions is set to a region and its center is determined [29]. Secondly, we get the distance between user's current location and their nearest check-in center, then we normalize the distance and convert it to a weight value. Thirdly, we inject different magnitudes of noise into the location information based on the weight value. The closer to the check-in center, the lower level the noise injected, since the closer to the check-in center, the more prosperous the place is, and the more users may check-in, the higher the population density is. Even if a small level of noise is added, it is difficult for an attacker to distinguish the target user. Motivated by Dou [30], the weighted noise injection method is shown in formula (9):

$$\mathrm{M}(\mathrm{I}_{u_i}) = f(\mathrm{I}_{u_i}) + Lap\left(\frac{\Delta \mathrm{f}}{e^{-w_{ui}*\varepsilon}}\right) \tag{9}$$

, where I_{ui} represents the real-time location of user u_i, and $W_{ui} \epsilon [0, 1]$ denotes the distance between the user's current location and his nearest check-in center.

Then, we provide the verification that formula (9) can keep differential privacy. To ensure that the algorithm's privacy budget ε is within the threshold while using the differential privacy technique multiple times, this paper needs to use the sequence combination property and parallel combination property to prove that the proposed algorithm satisfies the requirements of differential privacy. We inject the weighted Laplace noise into the query function $\mathrm{f}(\mathrm{I}_{u_i}) = \mathrm{f}(\mathrm{I}_{u_i1}) + \mathrm{f}(\mathrm{I}_{u_i2}) + \ldots + \mathrm{f}(\mathrm{I}_{u_it})^T$, and the output function is:

$$\begin{aligned}\mathrm{M}(\mathrm{I}_{u_i}) = \mathrm{f}(\mathrm{I}_{u_i}) + < Lap_1\left(\frac{\Delta f}{e^{-w_{ui}*\varepsilon}}\right) \\ + Lap_2\left(\frac{\Delta f}{e^{-w_{ui}*\varepsilon}}\right), \ldots, Lap_t\left(\frac{\Delta f}{e^{-w_{ui}*\varepsilon}}\right)\end{aligned} \tag{10}$$

, where $\Delta \mathrm{f} = |\ \mathrm{f}(I_{u_i}) - \mathrm{f}(\mathrm{I}'_{ui})\ |$, and the perturbed data is denoted as I'_{ui}. According to relevant definitions, the following formula is the condition that algorithm M satisfies differential privacy:

$$\Pr[\mathrm{M}(I_{ui}) = \mathrm{O}] < e^{\varepsilon *}\ \Pr[\mathrm{M}(\mathrm{I'_{ui}}) = \mathrm{O}] \tag{11}$$

And then, we will prove that formula (10) satisfies formula (11).

Assuming that $\mathrm{f}(\mathrm{I}_{u_i}) = (\mathrm{I}_{u_i1}, \mathrm{I}_{u_i2}, \ldots, \mathrm{I}_{u_it})^T$

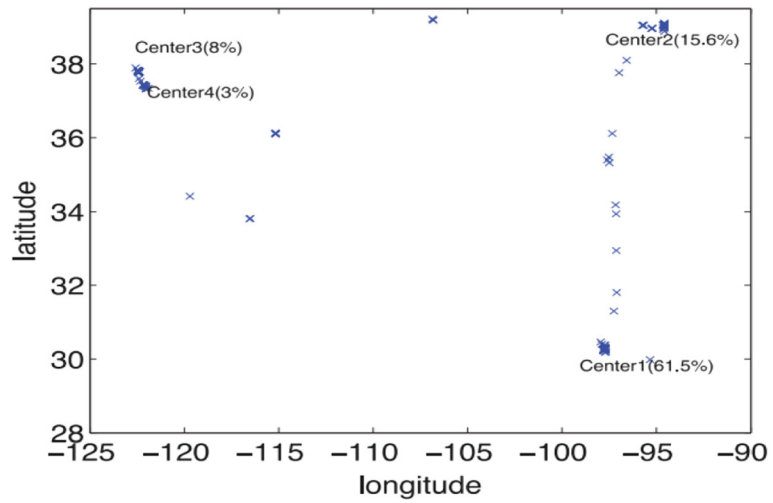

Fig. 2 Check-in Multi-centers

$$\begin{aligned} \mathrm{f}(I'_{ui}) &= (I'_{ui1}, I'_{ui2} \ldots + I'_{uit})^T \\ &= \left(I_{ui1} + \Delta \mathrm{I}_{ui1}, \mathrm{I}_{ui2} + \Delta \mathrm{I}_{ui2,}, \ldots, \mathrm{I}_{uit} + \Delta \mathrm{I}_{uit}\right) \end{aligned} \tag{12}$$

$$\Delta \mathrm{f} = \max_{I_{u_i}, I'_{u_i}} | \ \mathrm{f}\left(I_{u_i} - I'_{u_i}\right) | = \ \max | \sum_{j}^{t} | \ I_j - I \ |) = \ \max\left(\sum_{j}^{t} |\Delta I_i|\right) \tag{13}$$

The output variable is $\mathrm{O} = (\mathrm{y}_{u_i 1}, \mathrm{y}_{u_i 2}, \ldots, \mathrm{y}_{u_i t})^T$, and then,

$$\frac{\Pr[\mathrm{M}(I_{u_i}) = \mathrm{O}]}{\Pr\left[\mathrm{M}\left(\mathrm{I}'_{ui}\right) = \mathrm{O}\right]} = e^{-\frac{\varepsilon}{\Delta f} * \left(\left|\mathrm{y}_{u_i j} - \Delta I_{ui}\right| - \left|\mathrm{y}_{u_i j}\right|\right)} \tag{14}$$

For $\left|\mathrm{y}_{u_i j} - \Delta I_{ui}\right| - \left|\ \mathrm{y}_{u_i j}\ \right|$, by the definition of absolute value inequality, we can get the following formula:

$$\begin{aligned} &\sum_{1}^{t}\left(\left|\mathrm{y}_{u_i j} - \Delta I_{ui}\right| - \left|\mathrm{y}_{u_i j}\right|\right) \le \sum_{1}^{t} | \ \Delta u_i \\ &\quad | \ \le \max\left(\sum_{j}^{t} \Delta u_i\right) = \Delta \mathrm{f} \end{aligned} \tag{15}$$

Substituting the above inequality by

$$e^{\frac{\varepsilon}{\Delta f} * \sum_{1}^{t}\left(\left|\mathrm{y}_{u_i j} - \Delta I_{ui}\right| - \left|\mathrm{y}_{u_i j}\right|\right)} \tag{16}$$

Then we can get the following formula,

$$\frac{\Pr[\mathrm{M}(I_{u_i}) = \mathrm{O}]}{\Pr\left[\mathrm{M}\left(\mathrm{I}'_{ui}\right) = \mathrm{O}\right]} = e^{-\frac{\varepsilon}{\Delta f} * \Delta f} = e^{\varepsilon} \tag{17}$$

Therefore, each noise injection process satisfies the ε-differential privacy. Since the check-in information contains locations of many users, according to the parallel combination of differential privacy, the algorithm satisfies $\max(w * \varepsilon_i)$-differential privacy. Since max(w) = 1, the method satisfies ε-differential privacy.

Input: users' check-in location set C
check-in matrix
user set U
privacy budget ε
Output: perturbed data C'

1. **For** u_i in U **do**:
2. Extract u_i' check-in centers
3. Calculate the distance D_{ui} (current location, nearest center)
   Normalize D_{ui}, $D^*_{ui} = w_{ui} = \frac{D_{ui} - D_{min}}{D_{max} - D_{min}}$
4. Calculate the noise $Lap_1(\frac{\Delta f}{e^{-w_{ui} * \varepsilon}})$
   Inject noise to C_i
5. **End for**
6. **Return** C'

Recommending the next new POIs

As users would prefer to visiting the near POIs rather than the POIs far way, we can see that geographical factor affects users' visiting behavior. To incorporate the geographical influence into POI recommendation, we construct a circular region, the center of which is the user's current location at time t, and all the POIs in the circle are the candidates for the user's next movement. The user's previous sequential check-in POI record is represented as $<\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_t \mathrm{I}_{t+1}>$, and the candidate for user's next movement is represented as I_{et+1}, and then we can calculate the probability of $\mathrm{O} = <\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_t \mathrm{I}_{t+1}>$ for each POI in the region based on Forward

algorithm. By ranking the probability of $O = <I_1, I_2, \ldots, I_t, I_{t+1}>$ for each POI in this region, we can get Top k POIs for users' recommendation list. Given the parameter $\mu = \{\pi, T, P\}$ and the POI observation sequence $O_u = <I_1, I_2, \ldots, I_t, I_{t+1}>$, then the probability of $Pr(O_u|\mu)$ can be calculated by Forward algorithm. The Forward algorithm uses the forward vector $\alpha_t(i)$ that mentioned above to reduce the computational cost. Forward algorithm is to use the recursive relationship of formula (1) to get the forward vector in next step. And the required probability is given by

$$\Pr(O_u|\mu) = \sum_{i=1}^{N} \alpha_{t+1}(i) \tag{18}$$

, where

$$\alpha_1(j) = \pi_j P_J I_1, 1 < j < N \tag{19}$$

The specific process of the POI prediction algorithm is given as follows:

Input: $\mu=\{\pi, T, P\}$
POI set I
User set U
Circular region scale ΔD
Rank number k
user's sequential check-in POI record $I_1, I_2, \ldots, I_t$
Output: top k recommended new POIs

1. **For** u_i in U **do**:
2. get u_i^{ts} current position
construct circular region R
3. **For** POI_i in R **do**:
regard POI_i as I_{et+1}
Calculate the $\Pr(I_1, I_2, \ldots, I_t, I_{t+1}|\mu)$ by Forward algorithm
4. **End for**
5. rank POIs according to $\Pr(I_1, I_2, \ldots, I_t, I_{t+1}|\mu)$
6. **End for**
7. **Return** top k recommended new POIs

Experiment

In this part, we evaluate the following: (1) how is the proposed approach of POI recommendation in comparison with the baseline and other state-of-the-art recommendation techniques? (2) how do the scale of area and the number of latent states affect the model accuracy? (3) how is the performance of proposed model when providing privacy preserving?

Check-in data characteristics

We choose two large-scale datasets for experiments, Gowalla and Foursquare. Gowalla provides a public API for crawling information about all users, including check-in history record. The Foursquare dataset is provided by Yin [31]. Each check-in record contains three columns of information: user, POI, time. The specific location of each POI is related to longitude and latitude. Table 2 shows the basic statistics of two datasets:

Table 2 Statistics of two datasets

	#User	#POI	#Check-in
Foursquare	114,508	62,462	1,434,668
Gowalla	107,092	1,280,969	6,442,892

Evaluation metrics

In order to investigate the quality of the next new POI recommendation, we adapt the standard metric: *precision@k*. It shows the percentage of places in the next POI candidate list has been visited for each user. Formally, $R_u(k)$ denotes the k recommended new POIs in candidate list and V_u^{new} denotes the visited new places of user u.

$$\text{Precision@k} = \frac{1}{S}\sum_{u=1}^{s} \frac{R_u(k) \cap V_u^{new}}{k} \tag{20}$$

Evaluated methods

We compare our proposed approach with the following methods:

(1) CLB: Current location-based recommendation proposes a model that captures sequential influence and geographical influence. This model finds the user's current location and then recommends new POIs based on collaborative filtering [18].

(2) PMC: First-order Markov Chain considers sequential influence, it uses the time series of POIs visited by all users and the last place visited by the target user to make POI recommendation predictions [16].

(3) PRME: this approach combines two kind of distances for POI prediction, it models users' check-in sequences in hidden space, moreover, it considers user's preference and geographical influence [15].

In the experiments, for the purpose of evaluating the quality of different algorithms, we split the two datasets into two non-overlapping sets: for each user, the first 10 months check-in data is train set and the remaining 2 months check-in data is test set. The settings of the number of user's latent state are 5 and 8 for Foursquare and Gowalla, respectively. The circular region scale are 13 and 15 for Foursquare and Gowalla, respectively. We set the threshold $\theta = 0.02$ and the region radius $d = 15$ in greedy clustering algorithm to find all the check-in centers.

Comparison on accuracy

Table 3 shows the comparison results between our model and the baseline methods, the result shows that:

(1) Our proposed model has a better performance than CLB and FMC. In order to find the users with similar

Table 3 Performance comparison

Metrics	Gowalla				Foursquare			
	CLB	FMC	PRME	Our	CLB	FMC	PRME	Our
P@5	0.028	0.033	0.038	**0.041**	0.026	0.030	0.038	**0.040**
Improve	46.43%	24.24%	7.89%		53.85%	33.33%	5.26%	
P@10	0.023	0.027	0.035	**0.038**	0.020	0.024	0.033	**0.035**
Improve	65.22%	40.74%	8.57%		75.00%	45.83%	6.06%	
P@15	0.017	0.023	0.027	**0.030**	0.018	0.021	0.024	**0.028**
Improve	76.47%	30.43%	11.11%		55.56%	33.33%	16.67%	
P@20	0.015	0.016	0.021	**0.023**	0.013	0.014	0.019	**0.021**
Improve	53.33%	43.75%	9.52%		61.54%	50.00%	10.53%	

POI sequence record, CLB establishes a directed graph with users' POI check-in sequences, however, it is hard to solve the problem of data sparsity. One possible explanation for FMC's poor performance could be that it only considers the sequential influence of the last visited POI, however, FMC ignores the sequential influence of other recently visited POIs.

(2) PRME gets improved performance compared with CLB and FMC, since PRME develops a Metric Embedding algorithm to model the sequential transition of all visited POIs, moreover, it also captures geographical influence.

(3) Our proposed method consistently outperforms PRME, improving at least 7% and 5% over PRME for Gowalla and Foursquare respectively. According to our analysis, experimental results shows that the user's latent state in HMM is effective to capture users' actions and calculate users' POI transition probability, moreover, the geographical influence is beneficial for the next new POI recommendation.

Effect of parameters

Effect of the scale

In this section, we aim to investigate the parameter scale that affects the quality of the next new POI recommendation. The scale is the radius of the circular region. It determines the amount of surrounding buildings and POIs near the user. Figure 3 shows the precision under different values of scale on two data sets. The result indicates that our method gets its maximum precision when the scale is 13 for Foursquare dataset and 15 for Gowalla dataset. As the value of scale increases from 11, the accuracy increases at first; however, when the scale reaches a certain value, the precision of the recommendation system begins to decline. This is because

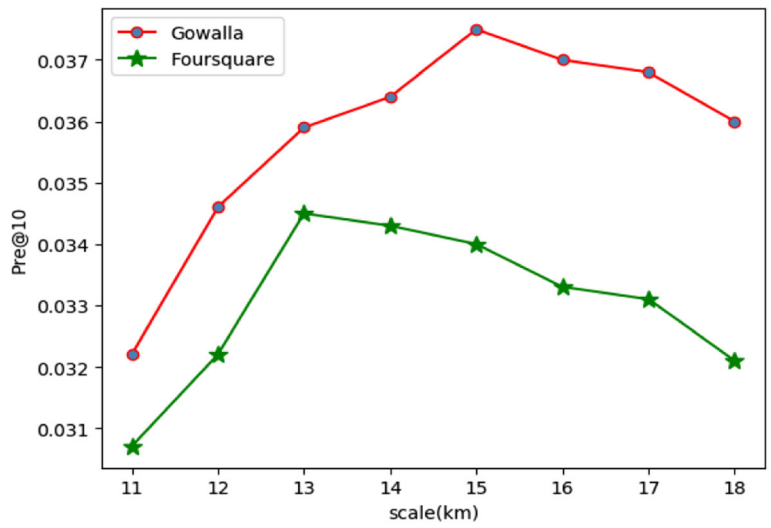

Fig. 3 Effect of scale

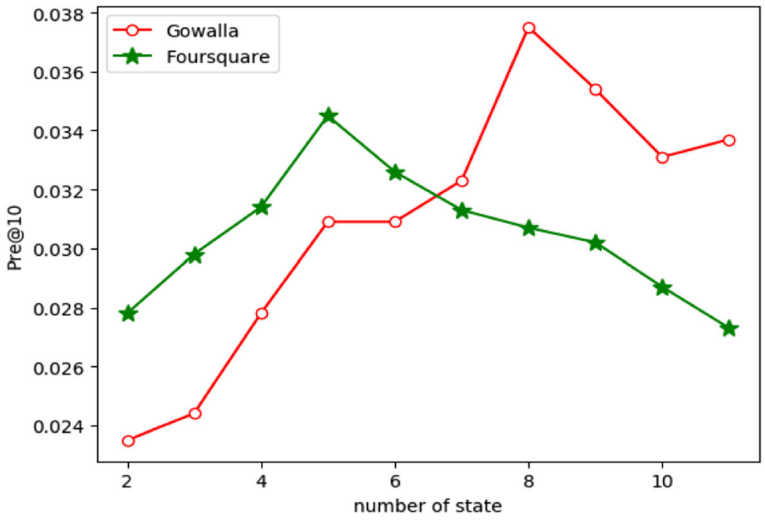

Fig. 4 Effect of user state number

when the scale is too small, the POIs in the area are too little to meet the needs of users; when the scale is too large, there are too many POIs in the area for users to choose, which also leads to lower accuracy.

Effect of the number of user state

In this section, we aim to investigate the number of user states that affects the quality of the next new POI recommendation. We describe the number of groups of user behavior over a period of time as the number of user states. In Fig. 4, the result indicates that when the number of user states is 5 for Foursquare and 8 for Gowalla, the approach gets the maximum precision. The number of a regular user's state is relatively small in real life, and a regular user state number provides generalization ability for this method.

Experiments on POI recommendation with privacy preserving

Comparison on perturbed data

In this section, we aim to compare our proposed privacy preserving algorithm based on the weight of distance(WD) with the traditional noise injection algorithm in terms of recommendation precision.

When the privacy parameter ε takes different values, the POI recommendation precision of the two methods is shown in Fig. 5. The privacy budget ε is an important parameter for determining the level of privacy preserving in differential privacy. By varying the privacy parameter ε from 0.1 to 1.0, we investigated the quality of our proposed algorithm at various privacy protection levels.

As the privacy budget ε increases, the precision of the POI recommendation increases. Figure 5 shows that our proposed algorithm performs better in POI

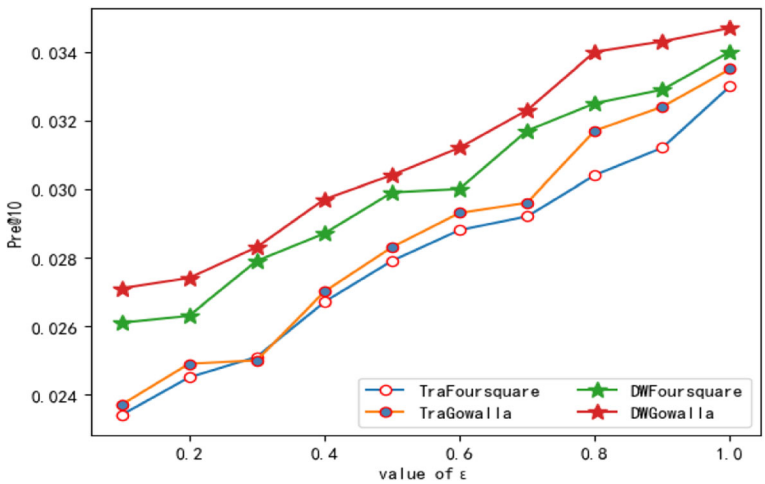

Fig. 5 Recommendation accuracy with perturbed data

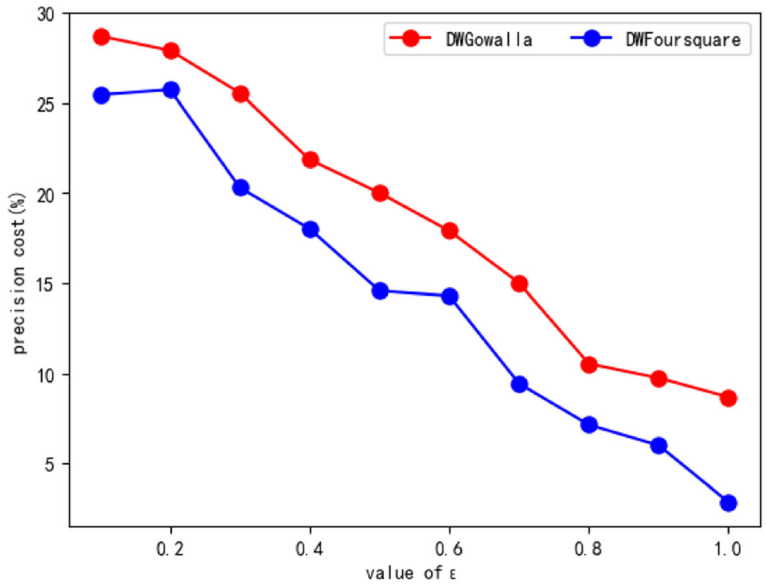

Fig. 6 Cost on precision

recommendation than traditional noise perturbation algorithm. In our proposed algorithm, different quantities of noise were injected into user's location based on the different distance between the user and the nearest check-in center. Malicious attackers can easily identify the current location of target user from the area with less POIs, therefore, the location with a long distance between the target user and the check-in center need to be added with a higher level of noise. As the quantity of noise injection increases, the data availability decreases. Our method has a better performance because our proposed noise injection method improves the availability of data.

Cost of privacy preserving

In this section, we aim to analyze what percentage of precision is lost as the cost of privacy preserving based on the weight of distance(WD). The precision of our proposed model without noise injection is 0.038 for Gowalla dataset and 0.035 for Foursquare dataset when $k = 10$. In our proposed POI recommendation with perturbed data, the precision changes as the privacy budget ε changes. Therefore, we analyze the cost of privacy preserving when the privacy budget ε takes different values.

Figure 6 shows how much percentage of precision is lost as the cost of privacy preserving by varying the privacy budget ε from 0.1 to 1.0. As the privacy budget ε increases, the cost of privacy preserving decreases. This is because when the privacy budget ε is larger, the level of privacy protection is lower, and the less noise is injected, and then the cost of privacy preserving is lower. When $\varepsilon = 0.1$, the cost of privacy preserving is 28.68%; when $\varepsilon = 1$, the cost of privacy preserving is 8.68%. Therefore, our proposed privacy preserving method protects users' location privacy with a little cost on precision.

Conclusion and future work

To address the problem of next new POI recommendation that supports privacy preserving, in this paper we propose a HMM-based POI sequential model to capture the successive check-in behavior by exploring the user's latent state for each user, and then we use EM algorithm to estimate the parameters of the model; we propose a check-in multi-center distance weighted noise injection method to protect users' location; finally, we combine geographical information and sequential information to provide new POI recommendation services that support privacy preserving for users. Performance evaluation conducted on two datasets shows that our proposed approach without noise injection improves the recommendation precision compared with other state-of-the art methods and meanwhile protects user's location information with little accuracy decrease.

For future work, we will try to consider different user states and periodic preference for users who have different lifestyles to further explore the capabilities of the next new POI recommendation.

Abbreviations

EM: Expectation Maximization; HMM: Hidden Markov Model; LBSNs: Location-Based Social Networks; MEC: Mobile edge computing; NFV: Network function virtualization; POI: Point-of-interest; WD: Weight of distance

Acknowledgements

We want to thank the authors of the literature cited in this paper for contributing useful ideas to this study.

Authors' contributions

Li Kuang, Shenmei Tu and Yangqi Zhang have written this paper and have done the research which supports it. Li Kuang, Shenmei Tu and Xiaoxian Yang has collaborated in the conception, research and design of the paper. All authors read and approved the final manuscript.

Author information

Li Kuang received her B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2004 and 2009 respectively. She is currently a professor at the school of Computer Science and Engineering, Central South University. Her research interests include Service Computing, Data Mining, Crowdsourcing software Ecosystem. She has authored or co-authored over 40 journal and conference publications.

Shenmei Tu received her bachelor's degrees in software engineering from Jiangxi Agricultural University, Nanchang, China, in 2017. She is currently pursuing a graduate student in software engineering at the school of Computer Science and Engineering, Central South University. Her research interests include service computing, data mining, recommendation system and privacy preserving.

Yangqi Zhang received his bachelor's degrees in software engineering from Central South University, Changsha, China, in 2019. He is currently pursuing a graduate student in software engineering at the school of Computer Science and Engineering, Central South University. His research interests include service computing, data mining and privacy preserving.

Xiaoxian Yang, received the Ph.D. degree in Management Science and Engineering from Shanghai University, Shanghai, China, in 2017. She is currently an assistant professor at Shanghai Polytechnic University, China. Her research interests include business process management, and formal method.

Funding

This work was funded by National Natural Science Foundation of China (No. 61772560), National Key R&D Program of China (No. 2018YFB1003800), National Natural Science Foundation of China (No. 61902236).

Availability of data and materials

The datasets supporting the conclusions of this article are available in http://net.pku.edu.cn/daim/hongzhi.yin/

Competing interests

The authors declare that there is no conflict of interest regarding the publication of this manuscript.

Author details

[1]School of Computer Science and Engineering, Central South University, Changsha, China. [2]School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China.

Received: 30 September 2019 Accepted: 22 January 2020
Published online: 07 February 2020

References

1. Chen Y, Deng S, Ma H, Yin J (2019) Deploying data-intensive applications with multiple services components on edge. Mob networks Appl:1–16
2. Gao H, Duan Y, Shao L, Sun X Transformation-based processing of typed resources for multimedia sources in the IoT environment. Wirel Networks:1–17
3. Kuang L, Yan X, Tan X et al (2019) Predicting taxi demand based on 3D convolutional neural network and multi-task learning. Remote Sens 11:1265
4. Liao Z, Zhao B, Liu S et al (2019) A prediction model of the project life-span in open source software ecosystem. Mob Networks Appl 24:1382–1391
5. Liao Z, Deng L, Fan X et al (2018) Empirical research on the evaluation model and method of sustainability of the open source ecosystem. Symmetry (Basel) 10:747
6. Zheng Y, Zhang L, Xie X, Ma W-Y (2009) Mining interesting locations and travel sequences from GPS trajectories. In: proceedings of the 18th international conference on world wide web. ACM, pp 791–800
7. Yuan Q, Cong G, Ma Z, et al (2013) Time-aware point-of-interest recommendation. In: proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 363–372
8. Chen C, Yin H, Yao J, Cui B (2013) Terec: a temporal recommender system over tweet stream. Proc VLDB Endow 6:1254–1257
9. Liu B, Fu Y, Yao Z, Xiong H (2013) Learning geographical preferences for point-of-interest recommendation. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 1043–1051
10. Ye M, Yin P, Lee W-C, Lee D-L (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, pp 325–334
11. Chen Y, Zhao J, Hu X, et al (2013) From interest to function: location estimation in social media. In: Twenty-Seventh AAAI Conference on Artificial Intelligence
12. Gao H, Tang J, Hu X, Liu H (2015) Content-aware point of interest recommendation on location-based social networks. In: Twenty-Ninth AAAI Conference on Artificial Intelligence
13. Ye M, Yin P, Lee W-C (2010) Location recommendation for location-based social networks. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 458–461
14. Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In: proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1082–1090
15. Feng S, Li X, Zeng Y, et al (2015) Personalized ranking metric embedding for next new POI recommendation
16. Cheng C, Yang H, Lyu MR, King I (2013) Where you like to go next: successive point-of-interest recommendation. In: Twenty-Third international joint conference on Artificial Intelligence
17. Feng S, Cong G, An B, Chee YM (2017) Poi2vec: geographical latent representation for predicting future visitors. In: Thirty-First AAAI Conference on Artificial Intelligence
18. Oppokhonov S, Park S, Ampomah IKE (2017) Current location-based next POI recommendation. In: Proceedings of the International Conference on Web Intelligence. ACM, pp 831–836
19. Sweeney L (2002) k-anonymity: a model for protecting privacy'International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10, 5 (2002) 557–570
20. Gedik B, Liu L (2004) A customizable k-anonymity model for protecting location privacy. Georgia Institute of Technology
21. Gentry C, Boneh D (2009) A fully homomorphic encryption scheme. Stanford University Stanford
22. Liu A, Wang W, Li Z et al (2017) A privacy-preserving framework for trust-oriented point-of-interest recommendation. IEEE Access 6:393–404
23. Duckham M, Kulik L (2005) A formal model of obfuscation and negotiation for location privacy. In: International conference on pervasive computing. Springer, pp 152–170
24. Ardagna CA, Cremonini M, di Vimercati SDC, Samarati P (2009) An obfuscation-based approach for protecting location privacy. IEEE Trans Dependable Secur Comput 8:13–27
25. Polat H, Du W (2003) Privacy-preserving collaborative filtering using randomized perturbation techniques. In: Third IEEE International Conference on Data Mining. IEEE, pp 625–628
26. Xiang Z, Deng S, Liu S et al (2016) Camer: a context-aware mobile service recommendation system. In: 2016 IEEE international conference on web services (ICWS). IEEE:292–299
27. Rabiner LR, Juang B-H (1986) An introduction to hidden Markov models. Ieee Assp Mag 3:4–16
28. Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Found trends®. Theor Comput Sci 9:211–407
29. Cheng C, Yang H, King I, Lyu MR (2012) Fused matrix factorization with geographical and social influence in location-based social networks. In: Twenty-Sixth AAAI Conference on Artificial Intelligence
30. Dou K, Guo B, Kuang L (2019) A privacy-preserving multimedia recommendation in the context of social network based on weighted noise injection. Multimed Tools Appl 78:26907–26926
31. Yin H, Wang W, Wang H et al (2017) Spatial-aware hierarchical collaborative deep learning for POI recommendation. IEEE Trans Knowl Data Eng 29: 2537–2551

Publisher's Note